

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BOARD OF PATENT APPEALS AND INTERFERENCES**

In re patent application of:  
Devins, et al.

Atty. Docket No.: BUR920010016US1

Serial No.: 10/060,750

Group Art Unit: 2123

Filed: January 30, 2002

Examiner: Sharon, Ayal I.

For: **SYSTEM FOR CONTROLLING EXTERNAL MODELS USED FOR  
VERIFICATION OF SYSTEM ON A CHIP (SOC) INTERFACES**

---

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPELLANTS' CORRECTED APPEAL BRIEF**

Sirs:

In response to the Notice of Non-Compliant Appeal Brief dated May 31, 2007, Appellant submits herewith a Corrected Appeal Brief in which Appellant respectfully appeals the final rejection of claims 2 and 8-34, in the Office Action dated February 14, 2006. A Notice of Appeal (and Pre-Appeal Brief Request for Review) was timely filed on May 12, 2006. A Notice of Panel Decision from Pre-Appeal Brief Review was mailed on March 21, 2007, which set forth a one month period for response. Therefore, Appellants' Appeal Brief is timely filed.

## Appeal Brief

### **I. REAL PARTY IN INTEREST**

The real party in interest is International Business Machines Corporation, Armonk, New York, assignee of 100% interest of the above-referenced patent application.

### **II. RELATED APPEALS AND INTERFERENCES**

There are no other appeals or interferences known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

### **III. STATUS OF CLAIMS**

Claims 2 and 8-34 are all the claims pending in the application and are set forth fully in the attached appendix (Section IX), are under appeal. Claims 1-34 were originally filed in the application. A non-final Office Action was issued on August 25, 2005 rejecting claims 1-34. The Appellants filed an Amendent under 37 C.F.R. §1.111 on November 23, 2005. A final Office Action was issued on February 14, 2006 rejecting claims 1-34. The Appellants filed an Amendent under 37 C.F.R. §1.116 on March 30, 2006 amending claims 2 and canceling claims 1, and 3-7. An Advisory Action was issued on May 2, 2006 indicating that the Amendent filed under 37 C.F.R. §1.116 on March 30, 2006 would not be entered. The Appellants filed a Notice of Appeal and Pre-Appeal Brief Request for Review timely on May 12, 2006.

A Notice of Panel Decision from Pre-Appeal Brief Review was issued on June 6, 2006, which indicated that the appeal proceed to the Board of Patent Appeals and Interferences. On January 17, 2007, a Notice of Abandonment was issued. Subsequently, on January 30, 2007, Appellants filed a Petition to Withdraw Holding of Abandonment Based on Failure to Receive Notice of Panel Decision. Appellants' petition was granted on February 26, 2007. A Notice of Panel Decision from Pre-Appeal Brief Review was mailed on March 21, 2007, which set forth a one month period for response.

## Appeal Brief

Claims 1-34 stand rejected under 35 U.S.C. §102(b) as being anticipated by Blaner, et al., “AN EMBEDDED PowerPC SOC for Test and Measurement Applications,” 13<sup>th</sup> Annual IEEE International ASIC/SOC Conference, 2000, September 13-16, 2000, pages 204-208, hereinafter referred to as Blaner. Claims 1-34 stand rejected under 35 U.S.C. §102(e) as being anticipated by Devins, et al. (U.S. Patent No. 6,487,699), hereinafter referred to as Devins. Appellants respectfully traverse these rejections based on the following discussion.

### **IV. STATEMENT OF AMENDMENTS**

A final Office Action dated February 14, 2006 stated all the pending claims 2, and 8-34 were rejected. The claims shown in the appendix (Section IX) are shown in their amended form as of the March 30, 2006 amendment.

### **V. SUMMARY OF CLAIMED SUBJECT MATTER**

The claimed invention provides a method, structure, and computer program product for a verification test bench system for testing an interface of a system-on-a-chip (SOC). One feature of the invention, as illustrated in FIG. 1 of Appellants’ disclosure, is an SOC 100 connected to a verification test bench 300. More specifically, the verification test bench system includes a verification interface model 210 connected to the SOC interface 101 and a test bench external bus interface unit (EBIU) 200 connected to the verification interface model 210 (p. 1, col. 2, para. 13; and, illustrated in FIG. 1). Independent claims 2, 8, and 15 define these features as follows: “a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface model”. Moreover, independent claims 21 and 28 define these features as follows: “connecting a verification interface model to said SOC interface; [and] connecting a test bench external bus interface unit (EBIU) to said verification interface model”.

Another feature of the invention is a test bench EBIU 200 connected to a SOC EBIU 205 within the SOC 100 (p. 1, col. 2, para. 13; and, illustrated in FIG. 1).

## Appeal Brief

Independent claims 2, 8, and 15 define these features as follows: “said test bench EBIU is connected to a SOC EBIU within said SOC”. Further, independent claims 21 and 28 define these features as follows: “connecting said test bench EBIU to a SOC EBIU within said SOC”.

Another feature of the invention is that the SOC EBIU 205 allows a test case running in the SOC 100 to control both the SOC interface 101 and the verification interface model 210 (p. 1, col. 2, para. 13; and, illustrated in FIG. 1). Independent claim 2 defines these features as follows: “said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model”.

Another feature of the invention is that the test bench EBIU 200 and the SOC EBIU 205 are mastered by the same processor in the SOC 100, such that the SOC interface 101 and the verification interface model 210 are programmed by the same test case running in the SOC (p. 1, col. 2, para. 13; and, illustrated in FIG. 1). The test case is written to utilize software drivers to configure a core or units that the test case is testing (p. 1, col. 2, para. 13; and, illustrated in FIG. 1). Independent claims 8 and 15 define these features as follows: “said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC”. Moreover, independent claim 15 defines these features as follows: “such that said SOC interface and said verification interface model are programmed by the same test case running in said SOC”.

Accordingly, as discussed in paragraph 0026 of Appellants’ disclosure, one advantage achieved with the invention is better software control. The invention allows the test case executing in the SOC 100 to use the same software driver, if appropriate, to program both interfaces 101, 210 (FIG. 1). The test case can also use one driver to program the SOC interface 101 and another to program the interface model 210, both of which are controlled by the test case. This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the external interface 210. The invention provides increased reusability and decreased development

## Appeal Brief

time because the invention uses the same or similar software written in the same language to program both interfaces.

### **VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

The issues presented for review by the Board of Patents Appeals and Interferences are whether claims 1-34 are unpatentable under 35 U.S.C. §102(b) as being anticipated by Blaner, and whether claims 1-34 are unpatentable under 35 U.S.C. §102(e) as being anticipated by Devins.

### **VII. ARGUMENT**

#### **A. The Rejections Based on Blaner**

##### **1. The Position in the Office Action**

In regards to Claim 1, Blaner teaches the following limitations:

1. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising:

a verification interface model connected to said SOC interface; and

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench” Emphasis added): “To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires in the testbench is connected to the external bus.”

a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to SOC EBIU within said SOC.

(See Blaner, especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II SOC Structure” Emphasis added): “ The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [process local bus] clock frequency. ... Further, the

## Appeal Brief

EBIU allows an off-chip device called the external bus master, to take ownership of the external bus and access attached memories.”

Moreover, Fig.2 of Blaner (see p.205), shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM, Flash, ROM, and ‘External Master’”.

While Blaner does not expressly teach that the “off-chip device, called the external bus master” also uses an EBIU in order to connect to the SOC’s EBIU, examiner finds this to be inherent because:

(1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc.), otherwise the interface does not work properly. This applies also to the external buses on SOC’s, and

(2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.” A complete system verification must also include a verification of the EBIU on the SOC, therefore the testbench must have an interface compatible with the SOC EBIU.

In regards to Claim 2, Blaner teaches the following limitations:

2. The verification test bench system in claim 1, wherein said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model.

(See Blaner, especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure”. Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.”

In regards to Claim 3, Blaner teaches the following limitations:

## Appeal Brief

3. The verification test bench system in claim 1, wherein said SOC interface and said verification interface model are programmed by a test case running in said SOC.

(See Blaner, especially: p.205, “II. SOC Structure and p208, “E. Verification Testbench”) Blaner teaches (See “II. SOC Structure”. Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at on-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories,”

Blaner also teaches (See “E. Verification Testbench”, Emphasis added): “Wrap backs are utilized as much as possible, and behavioral often contain hard-coded packet or stream data.”

Examiner interprets that “Wrap backs” do not containing functioning processors, and therefore must rely on the SOC processor.

In regards to Claim 4, Blaner teaches the following limitations:

4. The verification test bench in claim 3, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.

(See Blaner. especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure” Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.”

In regards to Claim 5, Blaner teaches the following limitations:

5. The verification test bench in claim 3, wherein said test case utilizes different software drivers to configure and control said SOC interface and said verification interface model.

(See Blaner especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II SOC Structure”, Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the

## Appeal Brief

EBIU allows an off-chip device called the external bus master, to take ownership of the external bus and access attached memories.”

In regards to Claim 6, Blaner teaches the following limitations:

6. The verification test bench system in claim 1, wherein said verification interface model tests an operational capability of said SOC interface.

(2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.”

In regards to Claim 7, Blaner teaches the following limitations:

7. The verification test bench system in claim 1, further comprising at least one additional verification interface model connected to said test bench EBIU for testing additional types of SOC interfaces.

(See Blaner, especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure”, Emphasis added): “... The external bus interface unit (EBIU controls up to eight banks types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.”

In regards to Claim 8, Blaner teaches the following limitations:

8. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising.

a verification interface model connected to said SOC interface; and

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench”. Emphasis added): “To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires in the testbench is connected to the external bus.”

a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and



## Appeal Brief

(See Blaner especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external access attached memories.”

Moreover, Fig.2 of Blaner (see p.205) shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM Flash, ROM, and ‘External Master’”.

While Blaner does not expressly teach that the “off-chip device, called the external bus master” also uses an EBIU in order to connect to the SOC’s EBIU, examiner finds this to be inherent because:

(1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc.), otherwise the interface does not work properly. This applies also to the external buses on SOC’s, and

(2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.” A complete system verification must also include a verification of the EBIU on the SOC, therefore the testbench must have an interface compatible with the SOC EBIU.

wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC.

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench”. Emphasis added): “Wrap backs are utilized as much as possible and behaviorals often contain hard-coded packet or stream data.”

## Appeal Brief

Examiner interprets that “Wrap backs” do not containing functioning processors, and therefore must rely on the SOC processor.

In regards to Claim 15, Blaner teaches the following limitations:

15. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC said verification test bench system comprising:

a verification interface model connected to said SOC interface; and

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench”. Emphasis added): “To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires the testbench is connected to the external bus.”

a test bench external bus Interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and

(See Blaner especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure. Emphasis added): “...The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master to take ownership of the external bus and access attached memories,”

Moreover, Fig.2 of Blaner (see p.205), shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM, Flash, ROM, and ‘External Master’”.

While Blaner does not expressly teach that the “off-chip device, called the external bus master also uses an EBIU in order to connect to the SOC’s EBIU, examiner finds this to be inherent because:

(1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc.), otherwise the interface does not work properly. This applies also to the external buses on SOC’s, and

## Appeal Brief

(2) Blaner teaches (See “E. Verification Testbench”). “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.” A complete system verification must also include a verification of the EBIU on the SOC, therefore the testbench must have an interface compatible with the SOC EBIU.

wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC, such that said SOC interface and said verification interface model are programmed by the same test case running in said SOC.

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench” Emphasis added): “Wrap backs are utilized as much as possible, and behaviorals often contain hard-coded packet or stream data.”

Examiner interprets that “Wrap backs” do not containing functioning processors, and therefore must rely on the SOC processor.

In regards to Claim 21, Blaner teaches the following limitations:

21. A method of testing a system-on-a-chip (SOC) interface of an SOC said method comprising:

connecting a verification interface model to said SOC interface;

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench”. Emphasis added): “To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires in the testbench is connected to the external bus.”

connecting a test bench external bus interface unit (EBIU) to said verification interface model; connecting said test bench EBI U to a SOC EBIU within said SOC; and comparing said SOC interface with said interface model.

(See Blaner, especially: p.205, “II. SOC Structure”)

## Appeal Brief

Blaner teaches (See “II. SOC Structure”. Emphasis added).”... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.”

Moreover, Fig.2 of Blaner (see p.205), shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM, Flash, ROM, and ‘External Master’”.

While Blaner does not expressly teach that the “off-chip device, called the external bus master” also uses an EBIU in order to connect to the SOC’s EBIU, examiner finds this to be inherent because:

(1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc.), otherwise the interface does not work properly. This applies also to the external buses on SOC’s, and

(2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.” A complete system verification must also include a verification of the EBIU on the SOC, therefore the testbench must have an interface compatible with the SOC EBIU.

In regards to Claim 28, Blaner teaches the following limitations:

28. A program storage device readable by machine tangibly embodying a program of instructions executable by the machine to perform a method for testing a system-on-a-chip (SOC) interface of an SOC, said method comprising:

connecting a verification interface model to said SOC interface;

(See Blaner, especially: p.208, “E. Verification Testbench”)

Blaner teaches (See “E. Verification Testbench”. Emphasis added): “To accomplish this synchronization, a memory-mapped external device containing

## Appeal Brief

software readable and writable registers that appear as wires in the testbench is connected to the external bus.”

connecting a test bench external bus interface unit (EBIU) to said verification interface model; connecting said test bench EBIU to a SOC EBIU within said SOC; and comparing said SOC interface with said interface model.

(See Blaner, especially: p.205, “II. SOC Structure”)

Blaner teaches (See “II. SOC Structure”. Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories”

Moreover, Fig.2 of Blaner (see p205), shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM, Flash, ROM and ‘External Master’”.

While Blaner does not expressly teach that tile “off-chip device, called the external bus master” also uses an EBIU in order to connect to the SOC’s EBIU, examiner finds this to be inherent because:

- (1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc otherwise the interface does not work properly. This applies also to the external buses on SOC’s, and
- (2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench.” A complete system verification must also include a verification of the EBIU on the SOC therefore the testbench must have an interlace compatible with the SOC EBIU.

Dependent Claims 2, 9, 16, 22 and 29 differ only in the limitations that they inherit from their parent claims. Therefore Claims 9, 16, 22, and 29 are rejected for the

## Appeal Brief

same reasons as claim 2, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 3, 10, 23, and 30 differ only in the limitations that they inherit from their parent claims. Therefore Claims 10, 23, and 30 are rejected for the same reasons as claim 3, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 4, 11, 17, 24, and 31 differ only in the limitations that they inherit from their parent claims. Therefore Claims 11, 17, 24, and 31 are rejected for the same reasons as claim 4, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 5, 12, 18, 25, and 32 differ only in the limitations that they inherit from their parent claims. Therefore Claims 12, 18, 25, and 32 are rejected for the same reasons as claim 5, in combination with the rejections of their respective parent claims, which are recited above,

Dependent Claims 6, 13, 19, 26, and 33 differ only in the limitations that they inherit from their parent claims. Therefore Claims 13, 19, 26, and 33 are rejected for the same reasons as claim 6, in combination with the rejections of their respective parent claims which are recited above.

Dependent Claims 7, 14, 20, 27, and 34 differ only in the limitations that they inherit from their parent claims. Therefore Claims 14, 20, 27, and 34 are rejected for the same reasons as claim 7, in combination with the rejections of their respective parent claims, which are recited above.

## **2. The Appellants' Position**

Appellants traverse the rejections because the prior art of record fails to teach or suggest the claimed features of “a verification interface model connected to said SOC interface” (independent claims 2, 8, and 15). As described above, the claimed invention connects a system-on-a-chip (SOC) to the verification test bench. Nothing within the prior art teaches connecting an SOC to a verification test bench, wherein both the SOC

## Appeal Brief

and the verification test bench are mastered by the same processor. Furthermore, nothing within the prior art teaches connecting an SOC to a verification test bench, wherein a test case running on the SOC controls both the SOC and the verification test bench. In the rejection, the Office Action attempts to broadly interpret the claim language and argues that some registers are equivalent to the SOC; however, the claims clearly provide for a “verification interface model connected to said SOC interface”. Therefore, as explained in greater detail below, Appellants respectfully submit that the prior art of record does not teach or suggest the claimed invention.

### a. Independent Claim 2

Appellants traverse the rejections because Blaner fails to teach the claimed feature of “a verification interface model connected to said SOC interface” (independent claim 2). The Office Action argues that Blaner has the ability to connect a *memory-mapped external device*, which contains software readable and writable registers that appear as wires in a testbench, to an external bus (Office Action, p. 3, section 10 (citing Blaner, p. 208, column 1, para. 2)). However, nothing in Blaner mentions connecting the testbench to an external SOC via the model interface and the SOC interface.

The “memory-mapped external device” of Blaner is not synonymous with the “verification interface model” of Appellants’ invention. Specifically, in Blaner, the “memory-mapped external device” is used to synchronize external activity to internal software. However, the “memory-mapped external device” of Blaner is not connected to the SOC interface. Instead, the “memory-mapped external device” of Blaner is “connected to the *external bus*” (Blander, p. 208, col. 1, para. 2 (emphasis added)). Conversely, as illustrated in FIG. 1 of Appellants’ disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100. The verification interface model 210 (which the Office Action asserts is taught by the “memory-mapped external device” of Blaner) is connected to the SOC interface 101. In Blaner, the “memory-mapped external device” is not connected to

## Appeal Brief

the SOC interface; rather, the "memory-mapped external device" is connected to "the external bus".

Furthermore, as described in paragraph 23 of Appellants' disclosure, the invention transfers data from the external verification interface model 210 to the SOC interface 101. The test case calls the software driver (SWD) 135-137 for the SOC interface 101 and instructs the software driver to configure the SOC interface 101 to receive data. Next, the test case calls the same SWD 135-137 and instructs the software driver to configure the external verification interface model 210 to send data. The SOC interface 101 and the external verification interface model 210 are implemented to respond to different unique addresses. Thus, when the test case calls the SWD 135-137 to perform some configuration on one of the interfaces, the test case sends the address of that interface along with the operation to be performed. The test case then sends test data to the unique data address of the external verification interface model 210. This data is sent from the SOC 100 through the SOC's external bus interface unit (EBIU) 205 to the external EBIU 200 and then along to the verification interface model 210. From there the data is sent through the verification interface model 210 into the SOC interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded. As discussed above, in Blaner, the "memory-mapped external device" (which the Office Action asserts teaches the verification interface model 210 of the claimed invention) is not connected to the SOC interface. Instead, the "memory-mapped external device" is connected to "the external bus". Whereas, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100.

Appellants further traverse the rejections because Blaner fails to teach the claimed feature wherein "said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model" (independent claim 2). The Office Action argues that the "external bus" and the "memory-mapped external device" of Blaner teach the "SOC interface 101" and the "verification interface model 210",



## Appeal Brief

respectively, of the claimed invention. However, Blaner does not include an SOC EBIU, or any other system component, that allows a test case running in the SOC to control both the “external bus” and the “memory-mapped external device”.

Instead, Blaner merely discloses that the SOC EBIU allows an *off-chip device* to take ownership of the “external bus” (Blaner, p. 205, col. 2, para. 3). First of all, Appellants submit that the off-chip device of Blaner is not allowed to take ownership of the “memory-mapped external device” (which the Office Action asserts teaches the verification interface model 210 of the claimed invention). Therefore, the off-chip device does not control *both* the “external bus” and the “memory-mapped external device”.

Further, Appellants submit that the fact that the off-chip device is allowed to take ownership of the external bus, does not allow a *test case running on the SOC* to control the external bus. The off-chip device is *external* to the SOC; thus, the off-chip device is not a test case running *on the SOC*. The fact that the off-chip device is allowed to take ownership of the external bus also does not allow a *test case running on the SOC* to control the “memory-mapped external device”.

Accordingly, Appellants submit that the “memory-mapped external device” of Blaner is connected to “the external bus” and is not connected to the SOC interface. Additionally, Applicants submit that Blaner does not teach or suggest that a *test case running in the SOC* can control *both* the SOC interface and the “memory-mapped external device”. Instead, Blaner merely discloses that an “off-chip device” can take ownership of the “external bus”. Therefore, Appellants respectfully submit that Blaner fails to teach or suggest the claimed feature of “a verification interface model connected to said SOC interface ... wherein said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model” as defined by independent claim 2.

### **b. Independent claim 8**

Appellants traverse the rejections because Blaner fails to teach the claimed feature of “a verification interface model connected to said SOC interface” (independent claim

## Appeal Brief

8). The Office Action argues that Blaner has the ability to connect a *memory-mapped external device*, which contains software readable and writable registers that appear as wires in a testbench, to an external bus (Office Action, p. 3, section 10 (citing Blaner, p. 208, column 1, para. 2)). However, nothing in Blaner mentions connecting the testbench to an external SOC via the model interface and the SOC interface.

The "memory-mapped external device" of Blaner is not synonymous with the "verification interface model" of Appellants' invention. Specifically, in Blaner, the "memory-mapped external device" is used to synchronize external activity to internal software. However, the "memory-mapped external device" of Blaner is not connected to the SOC interface. Instead, the "memory-mapped external device" of Blaner is "connected to the *external bus*" (Blander, p. 208, col. 1, para. 2 (emphasis added)). Conversely, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100. The verification interface model 210 (which the Office Action asserts is taught by the "memory-mapped external device" of Blaner) is connected to the SOC interface 101. In Blaner, the "memory-mapped external device" is not connected to the SOC interface; rather, the "memory-mapped external device" is connected to "the external bus".

Furthermore, as described in paragraph 23 of Appellants' disclosure, the invention transfers data from the external verification interface model 210 to the SOC interface 101. The test case calls the software driver (SWD) 135-137 for the SOC interface 101 and instructs the software driver to configure the SOC interface 101 to receive data. Next, the test case calls the same SWD 135-137 and instructs the software driver to configure the external verification interface model 210 to send data. The SOC interface 101 and the external verification interface model 210 are implemented to respond to different unique addresses. Thus, when the test case calls the SWD 135-137 to perform some configuration on one of the interfaces, the test case sends the address of that interface along with the operation to be performed. The test case then sends test data to the unique data address of the external verification interface model 210. This data is sent from the

## Appeal Brief

SOC 100 through the SOC's external bus interface unit (EBIU) 205 to the external EBIU 200 and then along to the verification interface model 210. From there the data is sent through the verification interface model 210 into the SOC interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded. As discussed above, in Blaner, the "memory-mapped external device" (which the Office Action asserts teaches the verification interface model 210 of the claimed invention) is not connected to the SOC interface. Instead, the "memory-mapped external device" is connected to "the external bus". Whereas, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100.

Accordingly, Appellants submit that the "memory-mapped external device" of Blaner is connected to "the external bus" and is not connected to the SOC interface. Therefore, Appellants respectfully submit that Blaner fails to teach or suggest the claimed feature of "a verification interface model connected to said SOC interface" as defined by independent claim 8.

### **c. Independent claim 15**

Appellants traverse the rejections because Blaner fails to teach the claimed feature of "a verification interface model connected to said SOC interface" (independent claim 15). The Office Action argues that Blaner has the ability to connect a *memory-mapped external device*, which contains software readable and writable registers that appear as wires in a testbench, to an external bus (Office Action, p. 3, section 10 (citing Blaner, p. 208, column 1, para. 2)). However, nothing in Blaner mentions connecting the testbench to an external SOC via the model interface and the SOC interface.

The "memory-mapped external device" of Blaner is not synonymous with the "verification interface model" of Appellants' invention. Specifically, in Blaner, the "memory-mapped external device" is used to synchronize external activity to internal software. However, the "memory-mapped external device" of Blaner is not connected to

## Appeal Brief

the SOC interface. Instead, the "memory-mapped external device" of Blaner is "connected to the *external bus*" (Blaner, p. 208, col. 1, para. 2 (emphasis added)). Conversely, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100. The verification interface model 210 (which the Office Action asserts is taught by the "memory-mapped external device" of Blaner) is connected to the SOC interface 101. In Blaner, the "memory-mapped external device" is not connected to the SOC interface; rather, the "memory-mapped external device" is connected to "the external bus".

Furthermore, as described in paragraph 23 of Appellants' disclosure, the invention transfers data from the external verification interface model 210 to the SOC interface 101. The test case calls the software driver (SWD) 135-137 for the SOC interface 101 and instructs the software driver to configure the SOC interface 101 to receive data. Next, the test case calls the same SWD 135-137 and instructs the software driver to configure the external verification interface model 210 to send data. The SOC interface 101 and the external verification interface model 210 are implemented to respond to different unique addresses. Thus, when the test case calls the SWD 135-137 to perform some configuration on one of the interfaces, the test case sends the address of that interface along with the operation to be performed. The test case then sends test data to the unique data address of the external verification interface model 210. This data is sent from the SOC 100 through the SOC's external bus interface unit (EBIU) 205 to the external EBIU 200 and then along to the verification interface model 210. From there the data is sent through the verification interface model 210 into the SOC interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded. As discussed above, in Blaner, the "memory-mapped external device" (which the Office Action asserts teaches the verification interface model 210 of the claimed invention) is not connected to the SOC interface. Instead, the "memory-mapped external device" is connected to "the external bus". Whereas, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit

## Appeal Brief

200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100.

In addition, the Office Action argues that Blaner discloses the claimed feature wherein “said SOC interface and said verification interface model are programmed by the same test case running in said SOC” as defined in independent claim 15 (Office Action, p. 8, para. 1 – p. 9, para. 1). In support for this contention, the Office Action references page 208 of Blaner, which discusses wrap backs. Specifically, the Office Action asserts that wrap backs are utilized as much as possible, and behaviorals often contain hard-coded packet or stream data. The Examiner interprets that wrap backs do not containing functioning processors, and therefore must rely on the SOC processor (Office Action, p. 9, para. 1).

Appellants submit that whether or not wrap backs rely on the SOC processor has nothing to do with a test case running on the SOC and nothing to do with programming the SOC interface and the “memory-mapped external device” (which the Office Action asserts teaches the verification interface model 210 of the claimed invention). Specifically, the wrap backs do not include a test case, or any other system component, running on the SOC. Moreover, nothing within Blaner mentions what programs the SOC interface or what programs the “memory-mapped external device”. Therefore, Blaner fails to disclose a system component that can program *both* the SOC interface and the “memory-mapped external device”. Instead, Blaner merely discloses that behaviorals often contain hard-coded packet or stream data and that wrap backs must rely on the SOC processor.

Therefore, Appellants respectfully submit that Blaner fails to teach or suggest the claimed feature of “a verification interface model connected to said SOC interface ... wherein said SOC interface and said verification interface model are programmed by the same test case running in said SOC” as defined by independent claim 15.

**d. Independent claims 21 and 28**

Appellants traverse the rejections because Blaner fails to teach the claimed feature of “connecting a verification interface model to said SOC interface” (independent claims 21 and 28). The Office Action argues that Blaner has the ability to connect a *memory-mapped external device*, which contains software readable and writable registers that appear as wires in a testbench, to an external bus (Office Action, p. 3, section 10 (citing Blaner, p. 208, column 1, para. 2)). However, nothing in Blaner mentions connecting the testbench to an external SOC via the model interface and the SOC interface.

The “memory-mapped external device” of Blaner is not synonymous with the “verification interface model” of Appellants’ invention. Specifically, in Blaner, the “memory-mapped external device” is used to synchronize external activity to internal software. However, the “memory-mapped external device” of Blaner is not connected to the SOC interface. Instead, the “memory-mapped external device” of Blaner is “connected to the *external bus*” (Blaner, p. 208, col. 1, para. 2 (emphasis added)). Conversely, as illustrated in FIG. 1 of Appellants’ disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100. The verification interface model 210 (which the Office Action asserts is taught by the “memory-mapped external device” of Blaner) is connected to the SOC interface 101. In Blaner, the “memory-mapped external device” is not connected to the SOC interface; rather, the “memory-mapped external device” is connected to “the external bus”.

Furthermore, as described in paragraph 23 of Appellants’ disclosure, the invention transfers data from the external verification interface model 210 to the SOC interface 101. The test case calls the software driver (SWD) 135-137 for the SOC interface 101 and instructs the software driver to configure the SOC interface 101 to receive data. Next, the test case calls the same SWD 135-137 and instructs the software driver to configure the external verification interface model 210 to send data. The SOC interface 101 and the external verification interface model 210 are implemented to respond to different unique addresses. Thus, when the test case calls the SWD 135-137 to perform some

## Appeal Brief

configuration on one of the interfaces, the test case sends the address of that interface along with the operation to be performed. The test case then sends test data to the unique data address of the external verification interface model 210. This data is sent from the SOC 100 through the SOC's external bus interface unit (EBIU) 205 to the external EBIU 200 and then along to the verification interface model 210. From there the data is sent through the verification interface model 210 into the SOC interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded. As discussed above, in *Blaner*, the "memory-mapped external device" (which the Office Action asserts teaches the verification interface model 210 of the claimed invention) is not connected to the SOC interface. Instead, the "memory-mapped external device" is connected to "the external bus". Whereas, as illustrated in FIG. 1 of Appellants' disclosure, the *external bus* interface unit 200 of the verification test bench 300 is connected to the *external bus* interface unit 205 of the SOC 100.

Accordingly, Appellants submit that the "memory-mapped external device" of *Blaner* is connected to "the external bus" and is not connected to the SOC interface. Therefore, Appellants respectfully submit that *Blaner* fails to teach or suggest the claimed feature of "connecting a verification interface model to said SOC interface" as defined by independent claims 21 and 28.

### **e. Dependent Claims 9, 16, 22, and 29**

Appellants traverse the rejections because *Blaner* fails to teach the claimed features wherein the SOC EBIU allows a test case running in the SOC to control both the SOC interface and the verification interface model (dependent claims 9, 16, 22, and 29). The Office Action argues that *Blaner* has the ability to use an EBIU to control up to eight banks of mixed types of memories and to operate at one-half the PLB clock frequency (Office Action, pp. 3-4, section 10 (citing p. 205, column 2, para. 3)). Further, the Office Action asserts that *Blaner* explains that the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.

## Appeal Brief

However, nothing in Blaner mentions a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface. As described in paragraph 24 of Appellants' disclosure, the invention allows the test case executing in the SOC 100 to use the same software driver, if appropriate, to program both interfaces 101, 210 (to configure and control the SOC interface and the verification interface model). The test case can also use one driver to program the SOC interface 101 and another to program the interface model 210, both of which are controlled by the test case (a test case running in the SOC to control both the SOC interface and the verification interface model). This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the external interface 210. The invention provides increased reusability and decreased development time because the invention uses the same or similar software written in the same language to program both interfaces (the same software driver to configure and control the SOC interface and the verification interface model).

Furthermore, as described in paragraph 26 of Appellants' disclosure, the invention represents a clean way of controlling external interfaces without the need for a complex control mechanism such as the conventional semaphore derived scheme used to enable communication between the SOC being tested and the external interface. The external bus mastering of the test bench EBIU 200 allows external model programming from the SOC test case. Thus, the same test case directly controls operations of the SOC interface 101 and the external model 210 (a test case running in the SOC to control both the SOC interface and the verification interface model).

Nothing within Blaner teaches the foregoing features of Appellants' invention, namely a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface. The fact that Blaner discloses, on page 205, column 2, paragraph 3, an external bus master for taking ownership of the external bus and accessing attached memories has nothing to do with programming both the SOC interface and the model interface with the same software driver. Neither the external bus



## Appeal Brief

master nor the external bus is utilized to program the SOC interface. Moreover, neither the external bus master nor the external bus is utilized to program the model interface. Neither the external bus master nor the external bus includes a software driver that is utilized to program any system components.

Blaner broadly discloses a design for an SOC and briefly discusses how it was tested. Blaner includes many details of how code was structured for re-useability across the many SOC designs and how simulation is accelerated by modeling the processor in the RTX code, rather than using the HDL model of the processor. However, Blaner specifically does not disclose building a testbench to stimulate/monitor the chip's I/O. Therefore, it is Appellants' position that Blaner fails to teach the claimed features wherein the SOC EBIU allows a test case running in the SOC to control both the SOC interface and the verification interface model (dependent claims 9, 16, 22, and 29).

### **f. Dependent Claims 10, 23, and 30**

The Office Action argues that Blaner discloses the claimed feature wherein "said SOC interface and said verification interface model are programmed by the same test case running in said SOC" as defined in dependent claim 10 and "programming said SOC interface and said verification interface model by a test case running in said SOC" as defined in dependent claims 23 and 30 (Office Action, p. 8, para. 1 – p. 9, para. 1). In support for this contention, the Office Action references page 208 of Blaner, which discusses wrap backs. Specifically, the Office Action asserts that wrap backs are utilized as much as possible, and behaviorals often contain hard-coded packet or stream data. The Examiner interprets that wrap backs do not containing functioning processors, and therefore must rely on the SOC processor (Office Action, p. 9, para. 1).

Appellants submit that whether or not wrap backs rely on the SOC processor has nothing to do with a test case running on the SOC and nothing to do with programming the SOC interface and the "memory-mapped external device" (which the Office Action asserts teaches the verification interface model 210 of the claimed invention). Specifically, the wrap backs do not include a test case, or any other system component,

## Appeal Brief

running on the SOC. Moreover, nothing within Blaner mentions what programs the SOC interface or what programs the “memory-mapped external device”. Therefore, Blaner fails to disclose a system component that can program *both* the SOC interface and the “memory-mapped external device”. Instead, Blaner merely discloses that behavioral models often contain hard-coded packet or stream data and that wrap backs must rely on the SOC processor.

Therefore, Appellants respectfully submit that Blaner fails to teach or suggest the claimed feature wherein “said SOC interface and said verification interface model are programmed by the same test case running in said SOC” as defined in dependent claim 10 and “programming said SOC interface and said verification interface model by a test case running in said SOC” as defined in dependent claims 23 and 30.

### **g. Dependent Claims 11, 17, 24, and 31**

Appellants traverse the rejections because Blaner fails to teach the claimed feature wherein the test case utilizes the same software driver to configure and control the SOC interface and the verification interface model (dependent Claims 11, 17, 24, and 31). The Office Action argues that Blaner has the ability to use an EBIU to control up to eight banks of mixed types of memories and to operate at one-half the PLB clock frequency (Office Action, pp. 3-4, section 10). Further, the Office Action asserts that Blaner explains that the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories. However, nothing in Blaner mentions a test case in the SOC that can use the same software driver to configure and control both the SOC interface and the verification interface model.

To the contrary, as described in paragraph 24 of Appellants’ disclosure, the invention allows the test case executing in the SOC 100 to use the same software driver, if appropriate, to program both interfaces 101, 210. This is an improvement over the conventional situation where the test case running within the SOC 100 controls the SOC interface 101, and another software program (written in a bus functional language) controls the verification interface model 210. The claimed invention provides increased

## Appeal Brief

reusability and decreased development time because the system and method uses the same or similar software written in the same language to program both interfaces.

Furthermore, as described in paragraph 26 of Appellants' disclosure, the claimed invention represents a clean way of controlling external interfaces without the need for a complex control mechanism such as the conventional semaphore derived scheme used to enable communication between the SOC being tested and the external interface. The external bus mastering of the test bench EBIU 200 allows external model programming from the SOC test case. Thus, the same test case directly controls operations of the SOC interface 101 and the verification interface model 210.

Nothing in Blaner teaches the foregoing features of Appellants' invention, namely a test case in the SOC that can use the same software driver to configure and control both the SOC interface and the model interface. The fact that Blaner discloses, on page 205, column 2, paragraph 3, an external bus master for taking ownership of the external bus and accessing attached memories has nothing to do with configuring and controlling both the SOC interface and the verification interface model with the same software driver. Therefore, it is Appellants' position that Blaner fails to teach the claimed feature wherein the test case utilizes the same software driver to configure and control the SOC interface and the verification interface model (dependent Claims 11, 17, 24, and 31).

### **h. Dependent Claims 12, 18, 25, and 32**

It is Appellants' position that Blaner does not teach the features defined in independent claims 8 and 15 and similarly does not teach the features defined in dependent claims 12, 18, 25, and 32, which depend upon independent claims 8 and 15. In view the foregoing, the Board is respectfully requested to reconsider and withdraw these rejections.

### **i. Dependent Claims 13, 19, 26, and 33**

It is Appellants' position that Blaner does not teach the features defined in independent claims 8 and 15 and similarly does not teach the features defined in

## Appeal Brief

dependent claims 13, 19, 26, and 33, which depend upon independent claims 8 and 15. In view the foregoing, the Board is respectfully requested to reconsider and withdraw these rejections.

### **j. Dependent Claims 14, 20, 27, and 34**

Appellants traverse the rejections because Blaner fails to teach at least one additional verification interface model connected to the test bench EBIU for testing additional types of SOC interfaces (dependent claims 14, 20, 27, and 34). The Office Action argues that the “memory-mapped external device ... connected to the external bus” teaches the verification interface model of the claimed invention (Office Action, p. 3, item 10). However, nothing within Blaner teaches or suggests at least one additional memory-mapped external device that could be connected to the external bus for testing additional types of SOC interfaces. Instead, Blaner only discloses a single memory-mapped external device.

Furthermore, the Office Action argues that an external bus interface unit (EBIU) “allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories” (Office Action, p. 6, item 16). However, nothing within Blaner teaches or suggests at least one additional external bus master for testing additional types of SOC interfaces. Instead, Blaner only discloses a single external bus master.

To the contrary, as illustrated in FIG. 1 and as discussed in paragraph 0023 of Appellants’ disclosure, item 215 represents an extra external interface model that the invention can be optionally used to test more than one type of SOC interface. Items 135-137 represent different software drivers (SWD) for driving or configuring different interfaces of the SOC. A software driver is software written only for a specific hardware device like a printer or a specific interface of an SOC. The test cases are written to utilize software drivers to configure the core or units they are testing.

## Appeal Brief

Therefore, it is Appellants' position that Blaner does not teach the claimed feature of at least one additional verification interface model connected to the test bench EBIU for testing additional types of SOC interfaces (dependent claims 14, 20, 27, and 34).

### **B. The Rejections Based on Devins**

#### **1. The Position in the Office Action**

In regards to Claim 1, Devins teaches the following limitations:

1. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising:

a verification interface model connected to said SOC interface; and

a test bench external bus interface unit (EBIU) connected to said verification interface model,

wherein said test bench EBIU is connected to a SOC EBIU within said SOC.

The "external bus interface logic" in the 'device is coupled to said system-on-chip device via a chip-external bus" that is claimed in claim 18 of the issued Devins patent is enabled in the specification of that patent in Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8- This corresponds to the "test bench external bus interface unit (EBIU)" claimed in claims 1, 8, and 15 of the instant application.

While the issued patent does not expressly teach that the "system-on-chip" device also uses an EBIU in order to connect to the test unit's EBIU. Examiner finds this to be inherent because the two ends of an interface must match (e.g. electrical plug and socket, phone Jack and socket, parallel port plugs and sockets, etc.), otherwise the interface does not work properly. This reasoning is especially relevant because the issued patent teaches (See col.4, lines 15-20):

The external bus interface logic 202 is designed to direct signals received via connection 107 to the appropriate logical address, and to convert tile particular bus protocol received into art internally-used format applicable to the command decode logic 203.

## Appeal Brief

In regards to Claim 2, Devins teaches the following limitations:

2. The verification test bench system in claim 1, wherein said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model.

(See Devins patent, especially: Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8)

In regards to Claim 3, Devins teaches the following limitations:

3. The verification test bench system in claim 1, wherein said SOC interface and said verification interface model are programmed by a test case running in said SOC.

(See Devins patent, especially: Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8)

In regards to Claim 4, Devins teaches the following limitations:

4. The verification test bench in claim 3, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.

(See Devins patent, especially: Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8)

In regards to Claim 5, Devins teaches the following limitations:

5. The verification test bench in claim 3, wherein said test case utilizes different software drivers to configure and control said Soc interface and said verification interface model.

(See Devins patent, especially: Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8)

In regards to Claim 6, Devins teaches the following limitations:

6. The verification test bench system in claim 1, wherein said verification interface model tests an operational capability of said SOC interface.

(See Devins patent, especially: Fig.2, Item 202, and further in column 4, lines 15-20 and 38-40, as well as col.5, lines 5-8)

In regards to Claim 7, Devins teaches the following limitations:

## Appeal Brief

7. The verification test bench system in claim 1, further comprising at least one additional verification interface model connected to said test bench EBIU for testing additional types of SOC interfaces.

(See Devins patent, especially; Fig.2, Item 202, and further in column 4, Pines 15-20 and 38-40, as well as col.5, lines 5-8)

Independent Claims 8, 15, 21, and 28 are rejected on the same grounds as independent claim 1.

Dependent Claims 2, 9, 16, 22, and 29 differ only in the limitations that they inherit from their parent claims. Therefore Claims 9, 16, 22, and 29 are rejected for the same reasons as claim 2 in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 3, 10, 23, and 30 differ only in the limitations that they inherit from their parent claims. Therefore Claims 10, 23, and 30 are rejected for the same reasons as claim 3, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 4, 11, 17, 24 and 31 differ only in the limitations that they inherit from their parent claims. Therefore Claims 11, 17 24, and 31 are rejected for the same reasons as claim 4, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 5, 12, 18, 25, and 32 differ only in the limitations that they inherit from their parent claims. Therefore Claims 12, 18, 25, and 32 are rejected for the same reasons as claim 5, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 6, 13, 19, 26, and 33 differ only in the limitations that they inherit from their parent claims. Therefore Claims 13, 19,26, and 33 are rejected for the same reasons as claim 6, in combination with the rejections of their respective parent claims, which are recited above.

Dependent Claims 7, 14, 20, 27, and 34 differ only in the limitations that they inherit from their parent claims. Therefore Claims 14,20, 27, and 34 are rejected for the

## Appeal Brief

same reasons as claim 7, in combination with the rejections of their respective parent claims, which are recited above.

Appellant's arguments filed 11/23/2005 have been fully considered but they are not persuasive.

Regarding the Blaner reference, Appellants unpersuasively argue (see p.10 of Appellants' arguments filed 11/23/2005) that it does not disclose:

connecting the testbench to an external SOC via the SOC interface and the model interface (independent claims 1, 8, 15, 21, and 28).

In response, Examiner notes that Blaner expressly teaches the following (see p.208, Section E. "Verification Testbench". Emphasis added):

System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench . . .

Because testcases are self-checking, all external activity is synchronized to the internal software. To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires in the testbench is connected to the external bus. Verilog models accept the wires as triggers and respond with status on the wires.

The testbench also utilizes a memory-mapped Verilog device, known as the external messaging unit (EMU), for displaying time-stamped trace and status information to users via the simulation console.

Blaner also teaches (see p.207, Section IV "Design Verification". Emphasis added):

The SOC was verified using a software-based verification system. A system exerciser approach was used, which consists of a group of software test programs linked together by a group of software test programs linked together by a special verification operating system called the Test Operating System (TOS) [1]. TOS is capable of scheduling test programs for execution and multi-tasking



## Appeal Brief

operations, enabling concurrent core execution. This provides the appropriate interface and bus transactions required for exercising the chip.

Blaner also teaches (see p.205, “II. SOC Structure”, Emphasis added):

Blaner teaches (See II. SOC Structure”, Emphasis added): “... The external bus interface unit (EBIU) controls up to eight banks of mixed types of memories and operates at one-half the PLB [processor local bus] clock frequency. ... Further, the EBIU allows an off-chip device, called the external bus master, to take ownership of the external bus and access attached memories.”

Moreover, Fig.2 of Blaner (see p.205), shows an EBIU in the extreme upper-left corner of the SOC block diagram. This diagram shows that the SOC EBIU connects externally to “SRAM, Flash, ROM, and ‘External Master’”.

While Blaner does not expressly teach that the “off-chip device, called the external bus master” also uses an EBIU in order to connect to The SOC’s EBIU, examiner finds this to be inherent because:

(1) The two ends of an interface must match (e.g. electrical plug and socket, phone jack and socket, parallel port plugs and sockets, etc.) otherwise the interface does not work properly, This applies also to the external buses on SOC’s, and

(2) Blaner teaches (See “E. Verification Testbench”): “System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench,” A complete system verification must also include a verification of the EBIU on the SOC, therefore the testbench must have an interlace compatible with the SOC EBIU.

Examiner therefore strongly disagrees with the Appellants’ argument that “nothing in Blaner mentions connecting the testbench to an external SOC via the SOC interface and the model interface,” (see p.10 of Appellants’ arguments) In particular, the above cited sections of Blaner expressly teach:

## Appeal Brief

- a. connecting the test bench to an external SOC (“The SOC was verified using a software-based verification system,” Blaner, p.207, Section IV “Design Verification”).
- b. Connecting the test bench to an external SOC via an SOC interface (“Further, the EBIU allows an off-chip device called the external bus master, to take ownership of the external bus and access attached memories.” Blaner, p.205, “II, SOC Structure”).
- c. Connecting the testbench to an external SOC via the SOC interface and the model interface (“TOS is capable of scheduling test programs for execution and multi-tasking operations enabling concurrent core execution. This provides the appropriate intercom and bus transactions required for exercising the chip.” Blaner, p.207, Section IV “Design Verification”).

Regarding the Blaner reference, the Appellants also unpersuasively argue (see p.11 of Appellants’ arguments filed 11/23/2005) that it does not disclose:

a test case in the SOC that can use the same software driver to program, both the SOC interface and the model interface (dependent claims 4, 11, 17, 24, and 31).

In response, Examiner notes that in addition to the above cited sections of the Blaner reference, Blaner also teaches the following (see p.207, Section IV “Design Verification” Sub-section D “TOS Structure”, Emphasis added):

The TOS [Test Operating System] system consists of a set of self-checking test programs written in C and linked together to operate independently of one another ... The system is constructed hierarchically from the top down, as shown in Fig.3. ... The top layer is called the chip exerciser, the middle contains the set of test application programs, and the bottom is where device drivers reside.

One motivation behind this hierarchical layering is to promote software reusability; isolating chip-specific details in the exerciser enables reuse of application and driver code on other chips.

## Appeal Brief

Examiner also notes that Fig.3 on p.207 shows a hierarchy of “User Interface & Chip Exerciser”, “Test Application” “Device Driver”, and “Core”. The first paragraph of Section II “SOC Structure” on p.205 teaches that “PLB and OPB are two of the three IBM CoreConnect buses”, indicating that the cores are components on the SOC. The first paragraph of Section II.B “OPB Subsystem” on p.206 confirms this definition with the teaching that “The OPB interconnects lower-performance cores to the PLB through the PLB-OPB bridge and HSDMA...”

Examiner also disagrees with the Appellants’ argument that “nothing in Blaner mentions a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface,” (see p.12 of Appellants’ arguments). In particular, Figure 3, and its associated text in Sections II.D and II.E teach that the same test application is used to program both the device driver and the core. This encompasses the claimed “verification interface model” and “SOC interface”.

Moreover, as cited earlier in this Office Action, Blaner teaches in Section II.E that

System verification requires stimulus/expectation models or devices to be attached to the external interfaces of the chip. Such models are often implemented in a testbench

Because testcases are self-checking, all external activity is synchronized to the internal software. To accomplish this synchronization, a memory-mapped external device containing software readable and writable registers that appear as wires in the testbench is connected to the external bus. Verilog models accept the wires as triggers and respond with status on the wires.

The testbench also utilizes a memory-mapped Verilog device, known as the external messaging unit (EMU), for displaying time-stamped trace and status information to users via the simulation console.

Regarding the Devins reference, Appellants unpersuasively argue (see pp.10-11 of Appellants’ arguments filed 11/23/2005) that it does not disclose:

connecting the testbench to an external SOC via the SOC interface and the model interface (independent claims 1, 8, 15, 21, and 28) a test case in the SOC that can

## Appeal Brief

use the same software driver to program both the SOC interface and the model interface (dependent claims 4, 11, 17, 24, and 31).

In response, Examiner notes that Devins expressly teaches the following (see col.2, lines 15-43. Emphasis added):

However, inefficiencies in current verification methodologies exacerbate time pressures. For example, Soc designs typically interface with cores that are external to the design. Existing methods of including such external cores in a verification test of a Soc design typically entail having to create special test cases to control the external cores; such test cases typically do not communicate with test cases being applied internally to the SOC and therefore lack realism. Calls to built-in simulator functions to control external cores are also used. However, such an approach is simulator-dependent and therefore not portable across simulators.

A verification methodology is needed which addresses the problems noted in the foregoing, which represent factors extending time-to-market.

### SUMMARY OF THE INVENTION

The present invention provides a method for communicating with and controlling cores which are external to a soc design during verification of the design, which avoids the above-noted inefficiencies in existing verification methods. According to the method, an external memory-mapped test device (EMMTD) is coupled between a SOC design being tested in simulation, and cores external to the SOC design. The EMMTD is coupled to the SOC via a chip-external bus, and coupled to external cores or to the external interfaces of cores internal to the SOC, via an EMMTD bi-directional bus.

The EMMTD processes signals received over the chip external bus and applies them to an external core or to an internal core with an external interface coupled to the EMMTD bi-directional bus. Internal logic in the EMMTD provides for control and status monitoring of a core coupled to the EMMTD bi-directional bus by enabling functions including driving data on the bus, reading the current

## Appeal Brief

state of data on the bus, and capturing positive and negative edge transitions on the bus.

A test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to the SOC by using the EMMTD to perform such functions as initiating external core logic which drives test signals to an internal core, directly controlling an internal core via its external interface or determining the status of an external core.

The EMMTD may also be physically embodied in, for example, an FPGA (Field Programmable Gate Array) or an ASIC (Application Specific integrated Circuit) usable with real hardware.

Examiner therefore finds that the Devins reference teaches the claimed limitations.

### 2. The Appellants' Position

In the rejection, the Office Action argues that Devins discloses external bus interface logic coupled to a SOC device via a chip-external bus. However, Devins does not disclose *connecting the testbench to an external SOC via the SOC interface and the model interface* (independent claims 2, 8, 15, 21, and 28). Moreover, Devins does not disclose *a test case in the SOC that can use the same software driver to program both the SOC interface and the model interface* (dependent claims 11, 17, 24, and 31).

Furthermore, the Office Action argues that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. However, although Devins broadly discusses an SOC coupled to external components, Devins does not teach or suggest connecting the testbench to an external SOC via the SOC interface and the model interface. The EMMTD of Devins does not teach the SOC interface 101 and the verification interface model 210 of the claimed invention, wherein the verification interface model 210 is also connected to a verification test bench EBIU, which is connected to an EBIU on the SOC. Instead, Devins merely discloses an EMMTD that is

## Appeal Brief

coupled between the SOC and “cores external to the SOC design”. Therefore, as explained in greater detail below, Appellants respectfully submit that the prior art of record does not teach or suggest the claimed invention.

### **a. Independent claim 2**

Appellants traverse the rejections because Devins fails to teach the claimed feature of “a verification interface model connected to said SOC interface” (independent claim 2). The Office Action proposes that Devins has the ability to couple “external bus interface logic” of a device to the SOC device via a chip-external bus. However, the “external bus interface logic” of Devins is not synonymous with the “model interface” of Appellants’ invention; rather, as pointed out on pages 12-13, item 28 of the Office Action, the “external bus interface logic” of Devins “corresponds to the ‘test bench external bus interface unit (EBIU)’”.

Therefore, Appellants submit that because the Office Action asserts that the “external bus interface logic” of Devins corresponds to the verification test bench EBIU 200 of the claimed invention, the “external bus interface logic” of Devins (which the Office Action asserts is coupled to the SOC) does not teach the verification interface model 210 of the claimed invention. In other words, the “external bus interface logic” of Devins does not teach *both* the test bench EBIU 200 and the verification interface model 210.

As illustrated in FIG. 1 and as defined in independent claim 2, the verification interface model 210 is connected to the SOC interface 101; and, the test bench EBIU 200 is connected to the SOC EBIU 205. Conversely, Devins does not disclose a verification test bench system having a verification interface model 210 AND a test bench EBIU 200. Although the Office Action argues that the “external bus interface logic” of Devins teaches the verification test bench EBIU 200 of the claimed invention, the “external bus interface logic” of Devins is not connected to *both* the SOC EBIU *and* the SOC interface. Therefore, it is Appellants’ position that Devins fails to teach the claimed feature of “a verification interface model connected to said SOC interface; and a test bench external

## Appeal Brief

bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC” as defined by independent claim 2.

The Office Action also asserts that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. Further, the Office Action argues that Devins discloses that a test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to the SOC, by using the EMMTD to perform such functions as initiating external core logic which drives test signals to an internal core, directly controlling an internal core via its external interface, or determining the status of an external core. Although Devins broadly discusses an SOC coupled to external components, Devins does not teach or suggest connecting the testbench to an external SOC via the SOC interface and the model interface.

More specifically, as illustrated in FIG. 1 of Appellants’ disclosure, the SOC 100 is connected to the verification test bench 300 via the SOC interface 101 and the verification interface model 210. The verification interface model 210 is connected to the verification test bench EBIU 200, which is connected to the SOC EBIU 205. Thus, the SOC EBIU 205 allows a test case running in the SOC 100 to control both the SOC interface 101 and the verification interface model 210. Such features are not taught by Devins. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and “cores external to the SOC design”. The EMMTD does not include a verification interface model that is connected to both a SOC interface and a verification test bench EBIU. Instead, the EMMTD merely comprises a bi-directional bus that is utilized to connect the SOC to “external cores”. The “external cores” do not include a verification test bench EBIU connected to the SOC EBIU. Moreover, the “external cores” do not include a verification test bench EBIU that is also connected to the verification interface model, which is connected to the SOC interface.

## Appeal Brief

Further, as illustrated in FIG. 1 of Appellants' invention, the SOC 100 is connected to the verification test bench 300. This connection is made by connecting the SOC interface 101 (of the SOC 100) and the verification interface model 210 (of the verification test bench 300). Moreover, as described in paragraph 25 of Appellants' disclosure, "[i]n FIG. 1, the invention transfers data from the external interface model 210 to the SOC interface 101 ... the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded." Thus, Appellants respectfully submit that, unlike the claimed invention, Devins does not teach connecting the testbench to an external SOC via the SOC interface and the model interface, as defined in independent claim 2.

### **b. Independent claim 8**

Appellants traverse the rejections because Devins fails to teach the claimed feature of "a verification interface model connected to said SOC interface" (independent claim 8). The Office Action proposes that Devins has the ability to couple "external bus interface logic" of a device to the SOC device via a chip-external bus. However, the "external bus interface logic" of Devins is not synonymous with the "model interface" of Appellants' invention; rather, as pointed out on pages 12-13, item 28 of the Office Action, the "external bus interface logic" of Devins "corresponds to the 'test bench external bus interface unit (EBIU)'".

Therefore, Appellants submit that because the Office Action asserts that the "external bus interface logic" of Devins corresponds to the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins (which the Office Action asserts is coupled to the SOC) does not teach the verification interface model 210 of the claimed invention. In other words, the "external bus interface logic" of Devins does not teach *both* the test bench EBIU 200 and the verification interface model 210.



## Appeal Brief

As illustrated in FIG. 1 and as defined in independent claim 8, the verification interface model 210 is connected to the SOC interface 101; and, the test bench EBIU 200 is connected to the SOC EBIU 205. Conversely, Devins does not disclose a verification test bench system having a verification interface model 210 AND a test bench EBIU 200. Although the Office Action argues that the "external bus interface logic" of Devins teaches the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins is not connected to *both* the SOC EBIU *and* the SOC interface. Therefore, it is Appellants' position that Devins fails to teach the claimed feature of "a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC" as defined by independent claim 8.

The Office Action also asserts that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. Further, the Office Action argues that Devins discloses that a test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to the SOC, by using the EMMTD to perform such functions as initiating external core logic which drives test signals to an internal core, directly controlling an internal core via its external interface, or determining the status of an external core. Although Devins broadly discusses an SOC coupled to external components, Devins does not teach or suggest connecting the testbench to an external SOC via the SOC interface and the model interface.

More specifically, as illustrated in FIG. 1 of Appellants' disclosure, the SOC 100 is connected to the verification test bench 300 via the SOC interface 101 and the verification interface model 210. The verification interface model 210 is connected to the verification test bench EBIU 200, which is connected to the SOC EBIU 205. Thus, the SOC EBIU 205 allows a test case running in the SOC 100 to control both the SOC interface 101 and the verification interface model 210. Such features are not taught by

## Appeal Brief

Devins. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and “cores external to the SOC design”. The EMMTD does not include a verification interface model that is connected to both a SOC interface and a verification test bench EBIU. Instead, the EMMTD merely comprises a bi-directional bus that is utilized to connect the SOC to “external cores”. The “external cores” do not include a verification test bench EBIU connected to the SOC EBIU. Moreover, the “external cores” do not include a verification test bench EBIU that is also connected to the verification interface model, which is connected to the SOC interface.

Further, as illustrated in FIG. 1 of Appellants’ invention, the SOC 100 is connected to the verification test bench 300. This connection is made by connecting the SOC interface 101 (of the SOC 100) and the verification interface model 210 (of the verification test bench 300). Moreover, as described in paragraph 25 of Appellants’ disclosure, “[i]n FIG. 1, the invention transfers data from the external interface model 210 to the SOC interface 101 ... the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded.” Thus, Appellants respectfully submit that, unlike the claimed invention, Devins does not teach connecting the testbench to an external SOC via the SOC interface and the model interface, as defined in independent claim 8.

### **c. Independent claim 15**

Appellants traverse the rejections because Devins fails to teach the claimed feature of “a verification interface model connected to said SOC interface” (independent claim 15). The Office Action proposes that Devins has the ability to couple “external bus interface logic” of a device to the SOC device via a chip-external bus. However, the “external bus interface logic” of Devins is not synonymous with the “model interface” of Appellants’ invention; rather, as pointed out on pages 12-13, item 28 of the Office Action, the “external bus interface logic” of Devins “corresponds to the ‘test bench external bus interface unit (EBIU)’”.

## Appeal Brief

Therefore, Appellants submit that because the Office Action asserts that the "external bus interface logic" of Devins corresponds to the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins (which the Office Action asserts is coupled to the SOC) does not teach the verification interface model 210 of the claimed invention. In other words, the "external bus interface logic" of Devins does not teach *both* the test bench EBIU 200 and the verification interface model 210.

As illustrated in FIG. 1 and as defined in independent claim 15, the verification interface model 210 is connected to the SOC interface 101; and, the test bench EBIU 200 is connected to the SOC EBIU 205. Conversely, Devins does not disclose a verification test bench system having a verification interface model 210 AND a test bench EBIU 200. Although the Office Action argues that the "external bus interface logic" of Devins teaches the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins is not connected to *both* the SOC EBIU *and* the SOC interface. Therefore, it is Appellants' position that Devins fails to teach the claimed feature of "a verification interface model connected to said SOC interface; and a test bench external bus interface unit (EBIU) connected to said verification interface model, wherein said test bench EBIU is connected to a SOC EBIU within said SOC" as defined by independent claim 15.

The Office Action also asserts that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. Further, the Office Action argues that Devins discloses that a test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to the SOC, by using the EMMTD to perform such functions as initiating external core logic which drives test signals to an internal core, directly controlling an internal core via its external interface, or determining the status of an external core. Although Devins broadly discusses an SOC coupled to external components, Devins does not teach or

## Appeal Brief

suggest connecting the testbench to an external SOC via the SOC interface and the model interface.

More specifically, as illustrated in FIG. 1 of Appellants' disclosure, the SOC 100 is connected to the verification test bench 300 via the SOC interface 101 and the verification interface model 210. The verification interface model 210 is connected to the verification test bench EBIU 200, which is connected to the SOC EBIU 205. Thus, the SOC EBIU 205 allows a test case running in the SOC 100 to control both the SOC interface 101 and the verification interface model 210. Such features are not taught by Devins. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and "cores external to the SOC design". The EMMTD does not include a verification interface model that is connected to both a SOC interface and a verification test bench EBIU. Instead, the EMMTD merely comprises a bi-directional bus that is utilized to connect the SOC to "external cores". The "external cores" do not include a verification test bench EBIU connected to the SOC EBIU. Moreover, the "external cores" do not include a verification test bench EBIU that is also connected to the verification interface model, which is connected to the SOC interface.

Further, as illustrated in FIG. 1 of Appellants' invention, the SOC 100 is connected to the verification test bench 300. This connection is made by connecting the SOC interface 101 (of the SOC 100) and the verification interface model 210 (of the verification test bench 300). Moreover, as described in paragraph 25 of Appellants' disclosure, "[i]n FIG. 1, the invention transfers data from the external interface model 210 to the SOC interface 101 ... the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded." Thus, Appellants respectfully submit that, unlike the claimed invention, Devins does not teach connecting the testbench to an external SOC via the SOC interface and the model interface, as defined in independent claim 15.

The Office Action also relies upon Devins' external bus interface logic 202 to reject Appellants' claims towards the SOC interface and the verification interface model,

## Appeal Brief

which are programmed by the test case running in the SOC (i.e., independent claim 15). Specifically, the Office Action highlights Devins' external bus interface logic 202, which is designed to direct signals received via connection 107 to the appropriate logical address, and to convert the particular bus protocol received into an internally-used format applicable to the command decode logic 203 (column 4, lines 15-20 and 38-40; column 5, lines 5-8).

Once more, the features cited in the prior art reference have nothing to do with utilizing the same software driver to configure and control the SOC interface and the verification interface model. Further, Devins does not mention allowing a test case running in the SOC to control both the SOC interface and the verification interface model; nor does Devins mention programming both the SOC interface and the verification interface model by the same test case running in the SOC, as defined in independent claim 15.

Appellants submit that the “external bus interface logic 202” of Devins is not utilized to control or program the EMMTD (which the Office Action asserts teaches the SOC interface 101 and the verification model interface 210 of the claimed invention). As discussed above, the EMMTD of Devins does not teach the SOC interface 101 and the verification interface model 210 of the claimed invention, wherein the verification interface model 210 is connected to a verification test bench EBIU, which is connected to an EBIU on the SOC. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and “cores external to the SOC design”. Further, the “external bus interface logic 202” of Devins is not utilized to control or program the EMMTD. Instead, the “external bus interface logic 202” of Devins is utilized to convert bus protocol into a format applicable to a command decode logic.

### **d. Independent claims 21 and 28**

Appellants traverse the rejections because Devins fails to teach the claimed feature of “connecting a verification interface model to said SOC interface” (independent claims 21 and 28). The Office Action proposes that Devins has the ability to couple

## Appeal Brief

“external bus interface logic” of a device to the SOC device via a chip-external bus. However, the "external bus interface logic" of Devins is not synonymous with the "model interface" of Appellants' invention; rather, as pointed out on pages 12-13, item 28 of the Office Action, the "external bus interface logic" of Devins "corresponds to the 'test bench external bus interface unit (EBIU)'".

Therefore, Appellants submit that because the Office Action asserts that the "external bus interface logic" of Devins corresponds to the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins (which the Office Action asserts is coupled to the SOC) does not teach the verification interface model 210 of the claimed invention. In other words, the "external bus interface logic" of Devins does not teach *both* the test bench EBIU 200 and the verification interface model 210.

As illustrated in FIG. 1 and as defined in independent claims 21 and 28, the verification interface model 210 is connected to the SOC interface 101; and, the test bench EBIU 200 is connected to the SOC EBIU 205. Conversely, Devins does not disclose a verification test bench system having a verification interface model 210 AND a test bench EBIU 200. Although the Office Action argues that the "external bus interface logic" of Devins teaches the verification test bench EBIU 200 of the claimed invention, the "external bus interface logic" of Devins is not connected to *both* the SOC EBIU *and* the SOC interface. Therefore, it is Appellants' position that Devins fails to teach the claimed feature of “connecting a verification interface model to said SOC interface; connecting a test bench external bus interface unit (EBIU) to said verification interface model; connecting said test bench EBIU to a SOC EBIU within said SOC” as defined by independent claims 21 and 28.

The Office Action also asserts that Devins expressly teaches an external memory-mapped test device (EMMTD) that is coupled between a SOC design being tested in simulation, and cores external to the SOC design. Further, the Office Action argues that Devins discloses that a test case being executed for SOC verification by a simulated embedded processor in the SOC can communicate with and control elements external to

## Appeal Brief

the SOC, by using the EMMTD to perform such functions as initiating external core logic which drives test signals to an internal core, directly controlling an internal core via its external interface, or determining the status of an external core. Although Devins broadly discusses an SOC coupled to external components, Devins does not teach or suggest connecting the testbench to an external SOC via the SOC interface and the model interface.

More specifically, as illustrated in FIG. 1 of Appellants' disclosure, the SOC 100 is connected to the verification test bench 300 via the SOC interface 101 and the verification interface model 210. The verification interface model 210 is connected to the verification test bench EBIU 200, which is connected to the SOC EBIU 205. Thus, the SOC EBIU 205 allows a test case running in the SOC 100 to control both the SOC interface 101 and the verification interface model 210. Such features are not taught by Devins. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and "cores external to the SOC design". The EMMTD does not include a verification interface model that is connected to both a SOC interface and a verification test bench EBIU. Instead, the EMMTD merely comprises a bi-directional bus that is utilized to connect the SOC to "external cores". The "external cores" do not include a verification test bench EBIU connected to the SOC EBIU. Moreover, the "external cores" do not include a verification test bench EBIU that is also connected to the verification interface model, which is connected to the SOC interface.

Further, as illustrated in FIG. 1 of Appellants' invention, the SOC 100 is connected to the verification test bench 300. This connection is made by connecting the SOC interface 101 (of the SOC 100) and the verification interface model 210 (of the verification test bench 300). Moreover, as described in paragraph 25 of Appellants' disclosure, "[i]n FIG. 1, the invention transfers data from the external interface model 210 to the SOC interface 101 ... the data is sent through the interface model 210 into the SOC's interface 101 which is configured to receive data. Once the data is back in the SOC 100, the test case checks it for correctness and a test status is recorded." Thus, Appellants respectfully submit that, unlike the claimed invention, Devins does not teach

## Appeal Brief

connecting the testbench to an external SOC via the SOC interface and the model interface, as defined in independent claims 21 and 28.

### **e. Dependent Claims 9, 16, 22, and 29**

Appellants traverse the rejections because Devins fails to teach the claimed features of a SOC EBIU that allows a test case running in the SOC to control both the SOC interface and the verification interface model (dependent claims 9, 16, 22, and 29). The Office Action asserts that such features are taught by Devins' external bus interface logic 202 (column 4, lines 15-20 and 38-40; column 5, lines 5-8). Specifically, the Office Action highlights Devins' external bus interface logic 202, which is designed to direct signals received via connection 107 to the appropriate logical address, and to convert the particular bus protocol received into an internally-used format applicable to the command decode logic 203 (column 4, lines 15-20 and 38-40; column 5, lines 5-8).

However, Devins does not mention allowing a test case running in the SOC to control both the SOC interface and the verification interface model; nor does Devins mention programming both the SOC interface and the verification interface model by the test case running in the SOC. Therefore, it is Appellants' position that Devins fails to teach the claimed features of a SOC EBIU that allows a test case running in the SOC to control both the SOC interface and the verification interface model (dependent claims 9, 16, 22, and 29).

### **f. Dependent Claims 10, 23, and 30**

The Office Action also relies upon Devins' external bus interface logic 202 to reject Appellants' claims towards the SOC interface and the verification interface model, which are programmed by the test case running in the SOC (i.e., dependent claims 10, 23, and 30). Specifically, the Office Action highlights Devins' external bus interface logic 202, which is designed to direct signals received via connection 107 to the appropriate logical address, and to convert the particular bus protocol received into an internally-used format applicable to the command decode logic 203 (column 4, lines 15-20 and 38-40;



## Appeal Brief

column 5, lines 5-8).

Once more, the features cited in the prior art reference have nothing to do with utilizing the same software driver to configure and control the SOC interface and the verification interface model. Further, Devins does not mention allowing a test case running in the SOC to control both the SOC interface and the verification interface model; nor does Devins mention programming both the SOC interface and the verification interface model by the same test case running in the SOC, as defined in dependent claims 10, 23, and 30.

Appellants submit that the “external bus interface logic 202” of Devins is not utilized to control or program the EMMTD (which the Office Action asserts teaches the SOC interface 101 and the verification model interface 210 of the claimed invention). As discussed above, the EMMTD of Devins does not teach the SOC interface 101 and the verification interface model 210 of the claimed invention, wherein the verification interface model 210 is connected to a verification test bench EBIU, which is connected to an EBIU on the SOC. Instead, Devins merely discloses an EMMTD that is coupled between the SOC and “cores external to the SOC design”. Further, the “external bus interface logic 202” of Devins is not utilized to control or program the EMMTD. Instead, the “external bus interface logic 202” of Devins is utilized to convert bus protocol into a format applicable to a command decode logic.

### **g. Dependent Claims 11, 17, 24, and 31**

Appellants traverse the rejections because Devins fails to teach the claimed feature wherein the test case utilizes the same software driver to configure and control the SOC interface and the verification interface model (dependent Claims 11, 17, 24, and 31). The Office Action asserts that such features are taught by Devins' external bus interface logic 202 (column 4, lines 15-20 and 38-40; column 5, lines 5-8). Specifically, the Office Action highlights Devins' external bus interface logic 202, which is designed to direct signals received via connection 107 to the appropriate logical address, and to

## Appeal Brief

convert the particular bus protocol received into an internally-used format applicable to the command decode logic 203 (column 4, lines 15-20 and 38-40; column 5, lines 5-8).

However, the features cited in the prior art reference have nothing to do with utilizing the same software driver to configure and control the SOC interface and the verification interface model. Therefore, it is Appellants' position that Devins fails to teach the claimed feature wherein the test case utilizes the same software driver to configure and control the SOC interface and the verification interface model (dependent Claims 11, 17, 24, and 31).

### **h. Dependent Claims 12, 18, 25, and 32**

It is Appellants' position that Devins does not teach the features defined in independent claims 8 and 15 and similarly does not teach the features defined in dependent claims 12, 18, 25, and 32, which depend upon independent claims 8 and 15. In view the foregoing, the Board is respectfully requested to reconsider and withdraw these rejections.

### **i. Dependent Claims 13, 19, 26, and 33**

It is Appellants' position that Blaner does not teach the features defined in independent claims 8 and 15 and similarly does not teach the features defined in dependent claims 13, 19, 26, and 33, which depend upon independent claims 8 and 15. In view the foregoing, the Board is respectfully requested to reconsider and withdraw these rejections.

### **j. Dependent Claims 14, 20, 27, and 34**

Appellants traverse the rejections because Devins fails to teach at least one additional verification interface model connected to the test bench EBIU for testing additional types of SOC interfaces (dependent claims 14, 20, 27, and 34). The Office Action argues that the external bus interface logic 202 of Devins teaches the verification interface model of the claimed invention (Office Action, p. 14, item 34 (citing Devins,

## Appeal Brief

Fig. 2, Item 202; col. 4, lines 15-20; col. 5, lines 5-8)). However, nothing within Devins, including the portions cited by the Office Action, teaches or suggests at least one additional external bus interface logic 202 for testing additional types of SOC interfaces. Instead, Devins only discloses systems and methods which only include a single external bus interface logic 202.

To the contrary, as illustrated in FIG. 1 and as discussed in paragraph 0023 of Appellants' disclosure, item 215 represents an extra external interface model that the invention can be optionally used to test more than one type of SOC interface. Items 135-137 represent different software drivers (SWD) for driving or configuring different interfaces of the SOC. A software driver is software written only for a specific hardware device like a printer or a specific interface of an SOC. The test cases are written to utilize software drivers to configure the core or units they are testing.

In view of the foregoing, the Board is respectfully requested to reconsider and withdraw the rejections.

## VIII. CONCLUSION

In view of the foregoing, the Appellants respectfully submit that the collective cited prior art do not teach or suggest the features defined by independent claims 2, 8, 10, 21, and 28, and as such, claims 1-34 are patentable over Blaner alone or in combination with one another. Further, dependent claims 9, 11-20, 22-27, and 29-34 are similarly patentable over Devins alone or in combination with one another, not only by virtue of their dependency from patentable independent claims, respectively, but also by virtue of the additional features of the Appellants' claimed invention they define. Thus, the Appellants respectfully request that the Board reconsider and withdraw the rejections of claims 1-34 and pass these claims to issue.

Appeal Brief

Please charge any deficiencies and credit any overpayments to Attorney's Deposit  
Account Number 09-0456.

Respectfully submitted,

Date: June 26, 2007

/Duane N. Moore/  
Duane N. Moore, Esq.  
Registration No. 53,352

Gibb & Rahman, LLC  
2568-A Riva Road, Suite 304  
Annapolis, MD, 21401  
Voice: (410) 573-6501  
Fax: (301) 261-8825  
Customer No. 29154

**IX. CLAIMS APPENDIX**

1. (Canceled).
2. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising:
  - a verification interface model connected to said SOC interface; and
  - a test bench external bus interface unit (EBIU) connected to said verification interface model,wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model.
- 3-7. (Canceled).
8. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising:
  - a verification interface model connected to said SOC interface; and
  - a test bench external bus interface unit (EBIU) connected to said verification interface model,wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC.
9. The verification test bench system in claim 8, wherein said SOC EBIU allows a test case running in said SOC to control both said SOC interface and said verification interface model.

## Appeal Brief

10. The verification test bench system in claim 8, wherein said SOC interface and said verification interface model are programmed by the same test case running in said SOC.
11. The verification test bench in claim 10, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.
12. The verification test bench in claim 10, wherein said test case utilizes different software drivers to configure and control said SOC interface and said verification interface model.
13. The verification test bench system in claim 8, wherein said verification interface model tests an operational capability of said SOC interface.
14. The verification test bench system in claim 8, further comprising at least one additional verification interface model connected to said test bench EBIU for testing additional types of SOC interfaces.
15. A verification test bench system for testing a system-on-a-chip (SOC) interface of an SOC, said verification test bench system comprising:
  - a verification interface model connected to said SOC interface; and
  - a test bench external bus interface unit (EBIU) connected to said verification interface model,wherein said test bench EBIU is connected to a SOC EBIU within said SOC, and wherein said test bench EBIU and said SOC EBIU are mastered by the same processor in said SOC, such that said SOC interface and said verification interface model are programmed by the same test case running in said SOC.

## Appeal Brief

16. The verification test bench system in claim 15, wherein said SOC EBIU allows said test case to control both said SOC interface and said verification interface model.
17. The verification test bench in claim 15, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.
18. The verification test bench in claim 15, wherein said test case utilizes different software drivers to configure and control said SOC interface and said verification interface model.
19. The verification test bench system in claim 15, wherein said verification interface model tests an operational capability of said SOC interface.
20. The verification test bench system in claim 15, further comprising at least one additional verification interface model connected to said test bench EBIU for testing additional types of SOC interfaces.
21. A method of testing a system-on-a-chip (SOC) interface of an SOC, said method comprising:
  - connecting a verification interface model to said SOC interface;
  - connecting a test bench external bus interface unit (EBIU) to said verification interface model;
  - connecting said test bench EBIU to a SOC EBIU within said SOC; and
  - comparing said SOC interface with said interface model.
22. The method in claim 21, further comprising allowing, through said SOC EBIU, a test case running in said SOC to control both said SOC interface and said verification interface model.

## Appeal Brief

23. The method in claim 21, further comprising programming said SOC interface and said verification interface model by a test case running in said SOC.

24. The method in claim 23, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.

25. The method in claim 23, wherein said test case utilizes different software drivers to configure and control said SOC interface and said verification interface model.

26. The method in claim 21, wherein said comparing process tests an operational capability of said SOC interface.

27. The method in claim 21, further comprising:  
connecting at least one additional verification interface model to said test bench EBIU; and  
testing additional types of SOC interfaces.

28. A program storage device readable by machine tangibly embodying a program of instructions executable by the machine to perform a method for testing a system-on-a-chip (SOC) interface of an SOC, said method comprising:  
connecting a verification interface model to said SOC interface;  
connecting a test bench external bus interface unit (EBIU) to said verification interface model;  
connecting said test bench EBIU to a SOC EBIU within said SOC; and  
comparing said SOC interface with said interface model.



## Appeal Brief

29. The program storage device in claim 28, wherein said method further comprises allowing, through said SOC EBIU, a test case running in said SOC to control both said SOC interface and said verification interface model.

30. The program storage device in claim 28, wherein said method further comprises programming said SOC interface and said verification interface model by a test case running in said SOC.

31. The program storage device in claim 30, wherein said test case utilizes the same software driver to configure and control said SOC interface and said verification interface model.

32. The program storage device in claim 30, wherein said test case utilizes different software drivers to configure and control said SOC interface and said verification interface model.

33. The program storage device in claim 28, wherein said comparing process tests an operational capability of said SOC interface.

34. The program storage device in claim 28, wherein said method further comprises:  
connecting at least one additional verification interface model to said test bench EBIU; and  
testing additional types of SOC interfaces.

**X. EVIDENCE APPENDIX**

There is no other evidence known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**XI. RELATED PROCEEDINGS APPENDIX**

There is no other related proceedings known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.